PA1_template.rmd

Srikumar Gopal

July 20, 2016

Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

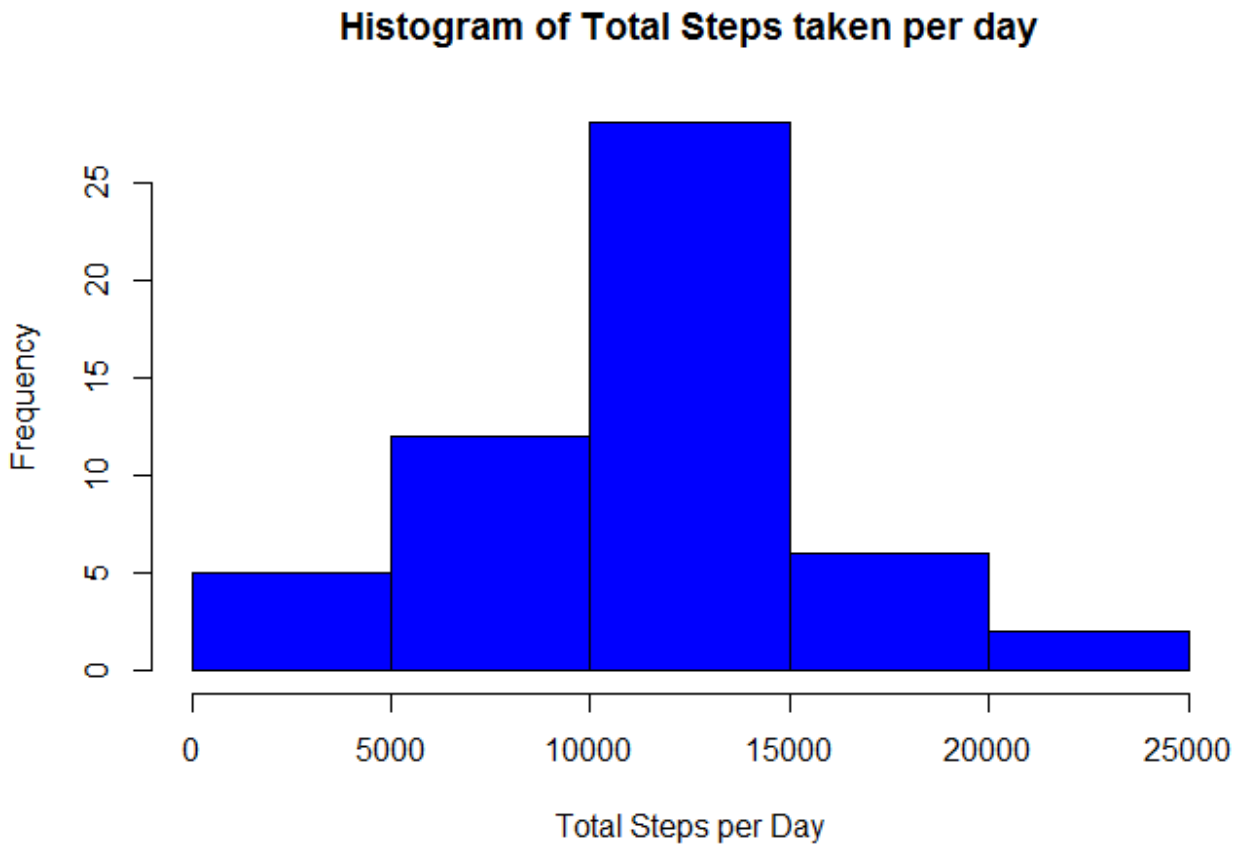
```
data <- read.csv("activity.csv")
data$date <- as.Date(data$date,"%Y-%m-%d")
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

What is mean total number of steps taken per day?

```
totsteps <- tapply(data$steps, data$date,sum)

hist(totsteps,col="blue",xlab="Total Steps per Day",
      ylab="Frequency", main="Histogram of Total Steps taken per day")
```




```
mean(totsteps,na.rm=TRUE)
```

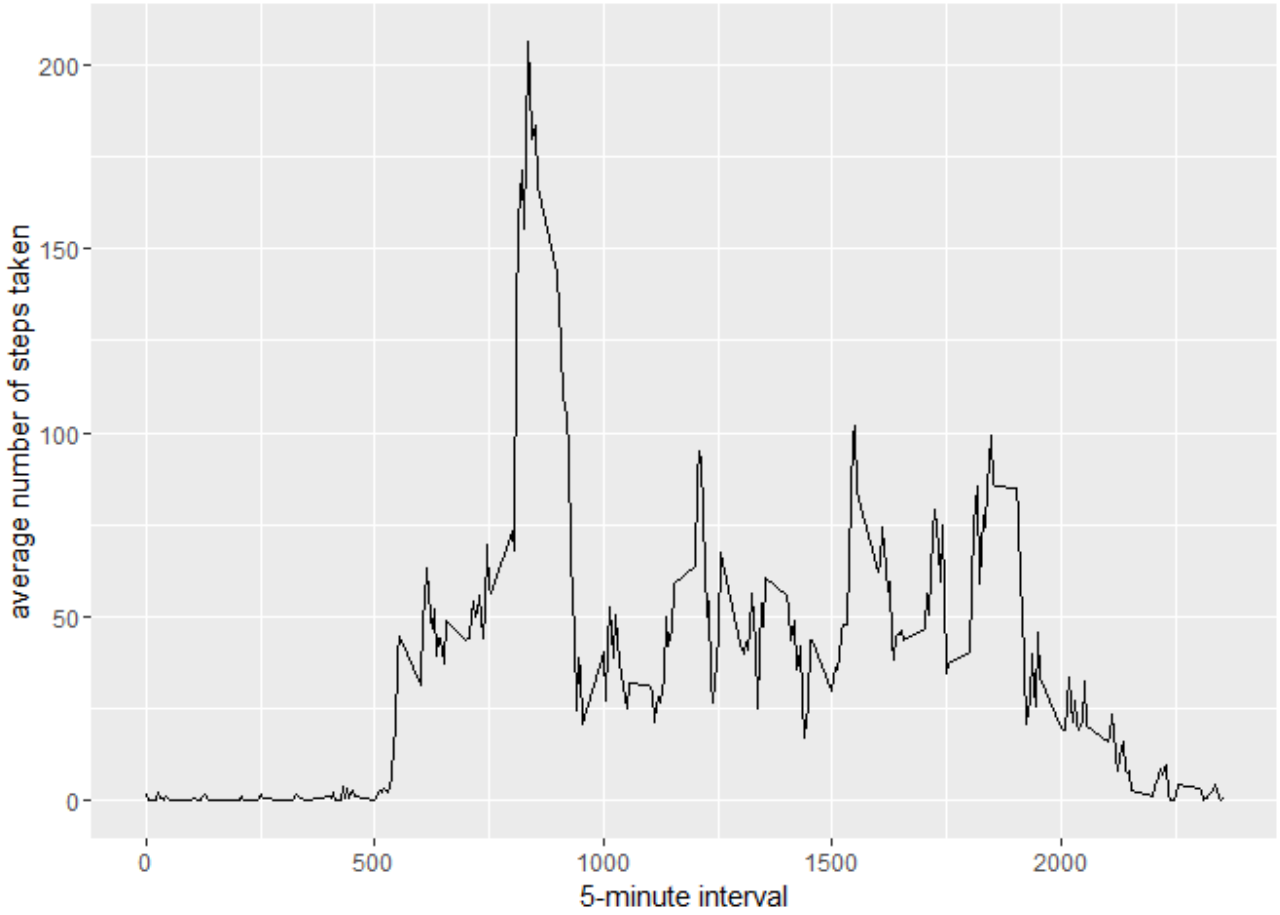
```
## [1] 10766.19
```

```
median(totsteps,na.rm=TRUE)
```

```
## [1] 10765
```

What is the average daily activity pattern?

```
library(ggplot2)
averages <- aggregate(x=list(steps=data$steps), by=list(interval=data$interval),
                      FUN=mean, na.rm=TRUE)
ggplot(data=averages, aes(x=interval, y=steps)) +
    geom_line() +
    xlab("5-minute interval") +
    ylab("average number of steps taken")
```



On average across all the days in the dataset, the 5-minute interval contains the maximum number of steps?

```
averages[which.max(averages$steps),]
```

```
##     interval    steps
## 104      835 206.1698
```

Inputing missing values

There are many days/intervals where there are missing values (coded as `NA`). The presence of missing days may introduce bias into some calculations or summaries of the data.

```
missing <- is.na(data$steps)
# How many missing
table(missing)
```

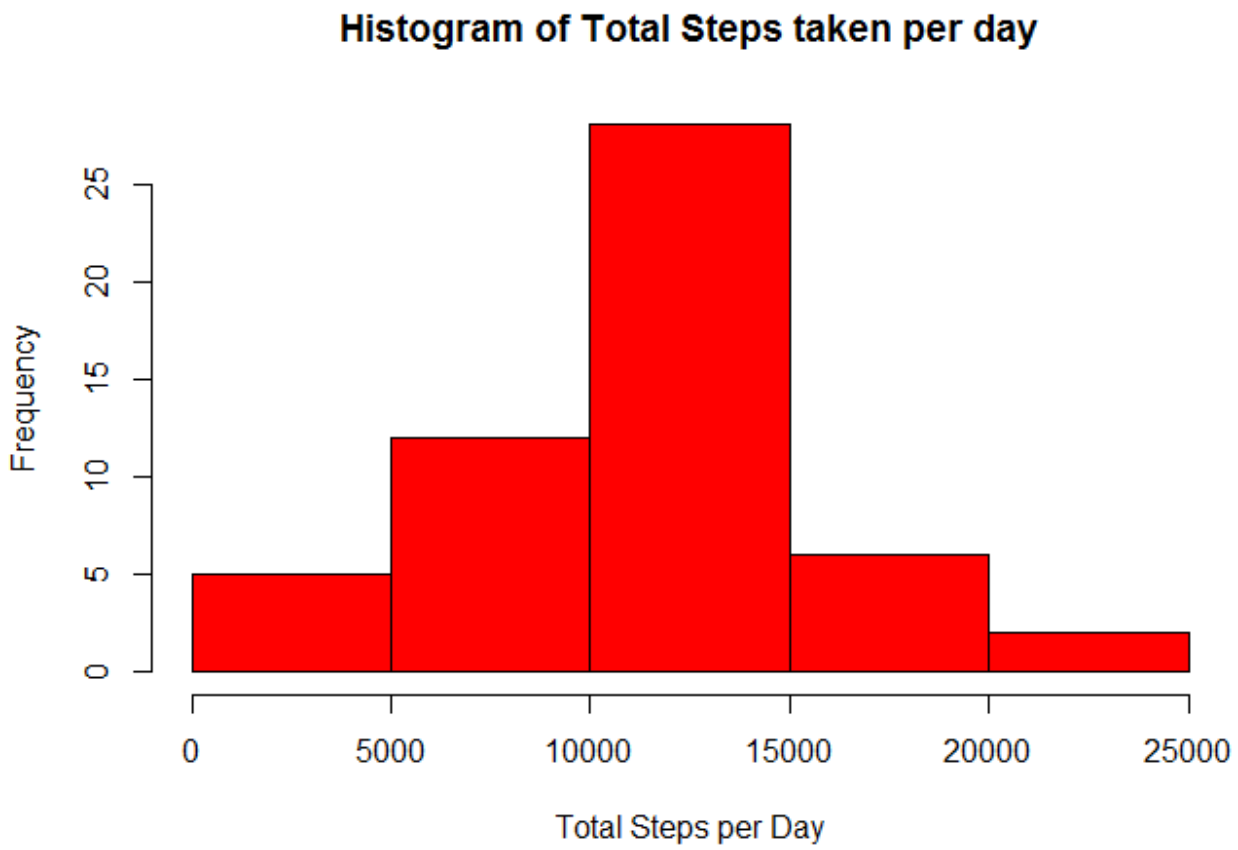
```
## missing
## FALSE  TRUE
## 15264  2304
```

All of the missing values are filled in with mean value for that 5-minute interval.

```
# Replace each missing value with the mean value of its 5-minute interval
fill.value <- function(steps, interval) {
    filled <- NA
    if (!is.na(steps))
        filled <- c(steps)
    else
        filled <- (averages[averages$interval==interval, "steps"])
    return(filled)
}
filled.data <- data
filled.data$steps <- mapply(fill.value, filled.data$steps, filled.data$interval)
```

Now, using the filled data set, let's make a histogram of the total number of steps taken each day and calculate the mean and median total number of steps.

```
totsteps <- tapply(data$steps, data$date, sum)
hist(totsteps, col = "red", xlab = "Total Steps per Day", ylab = "Frequency",
    main = "Histogram of Total Steps taken per day")
```



Are there differences in activity patterns between

weekdays and weekends?

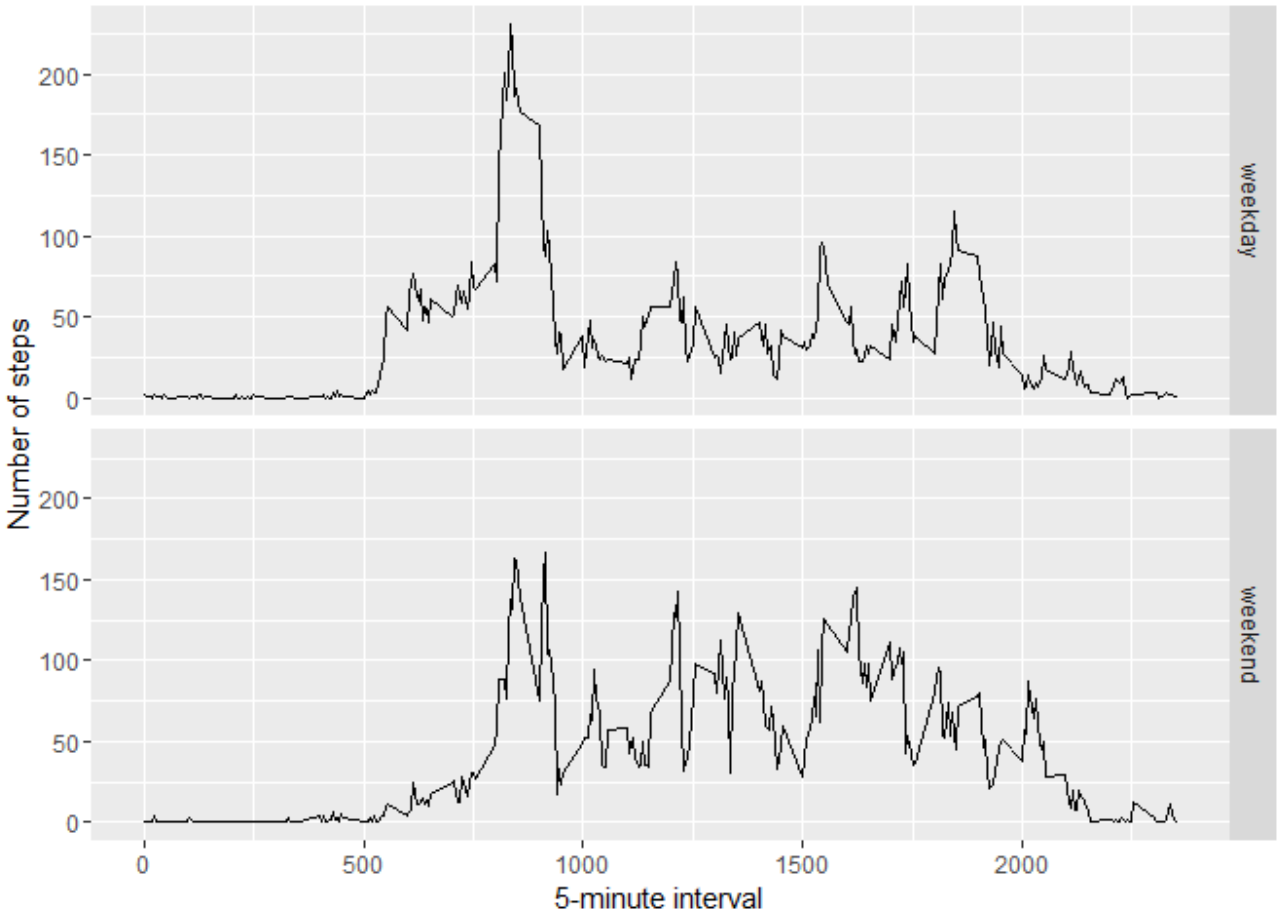
First, let's find the day of the week for each measurement in the dataset. In this part, we use the dataset with the filled-in values.

```
weekday.or.weekend <- function(date) {
    day <- weekdays(date)
    if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
        return("weekday")
    else if (day %in% c("Saturday", "Sunday"))
        return("weekend")
    else
        stop("invalid date")
}

filled.data$date <- as.Date(filled.data$date)
filled.data$day <- sapply(filled.data$date, FUN=weekday.or.weekend)
```

Now, let's make a panel plot containing plots of average number of steps taken on weekdays and weekends.

```
averages <- aggregate(steps ~ interval + day, data=filled.data, mean)
ggplot(averages, aes(interval, steps)) + geom_line() + facet_grid(day ~ .) +
    xlab("5-minute interval") + ylab("Number of steps")
```


Number of steps
200
150
100
50
0
weekday
weekend
0
500
1000
1500
2000
5-minute interval